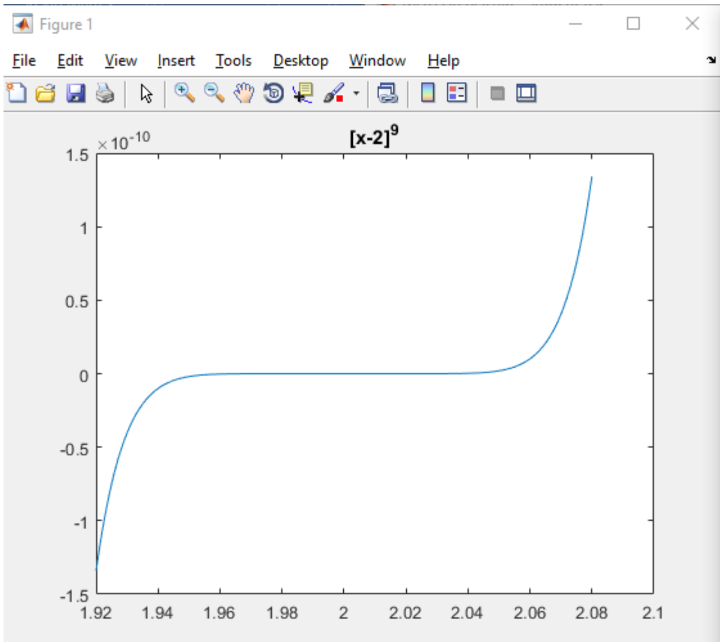
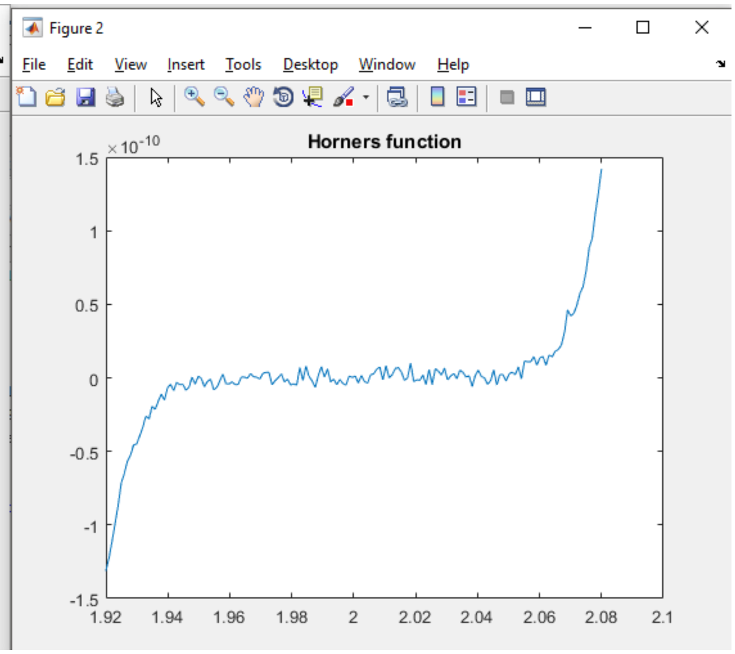
1





a) As seen on the graphs Function 2, Horner's method, is far less accurate. This is most likely because Horner's method is unstable and unreliable in terms of the positive or negative sign of the function.

b) Using the bisection method from the textbook, a root found was at 1.989067993164062 with 17 iterations. No values of r can be found such that |r − 2| < 10−6.

c) Because it's not feasible because the result is the tolerance given is too close too zero.

d) Applying fsolve with an initial guess of 1.9 still gives still gives a result of 1.900000000000.

2.

a) fsolve: [1.141e+01, -8.968e-01]

Newton solution: [5.00, 4.00]

46 iterations we're needed

b) fsolve: [7.719e-01 ,7.169e-02, 7.002e-01]

Newton solution: [1.667e+00, -6.667e-01, 1.333e+00]

57 iterations we're needed

c) fsolve: [-2.672986e-03 ,2.672986e-04, 4.073372e-04, 4.073372e-04]

Newton solution: [1, 2, 1, 1]

Iterations: Only 1 iteration was used

d) fsolve: [0,-3.162e-04]

Newton solution: [0,-Inf]

Only 2 iterations was used

e) fsolve: [1.004816e-02,1.005e-02]

Newton solution: [Inf,-Inf]

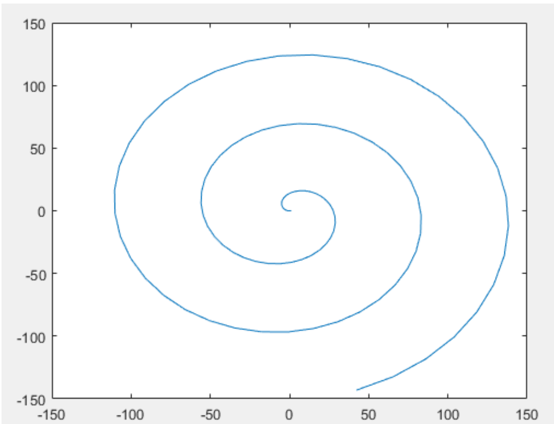
Only 2 iterations was used

In parts c , d and e the nonlinear solver fails to converge or converges to a point other than a solution.

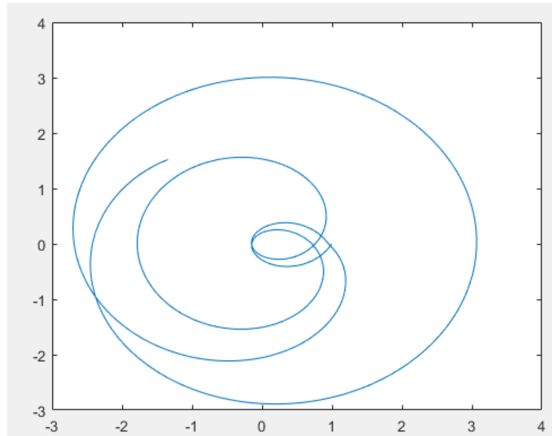
In all cases this is because the Jacobian of the system is singular, so the algorithm might converge to a point that is not a solution of the system of equations.

3. As seen in the graphs below, 10 000 steps are before the orbit appears to be qualitatively correct.

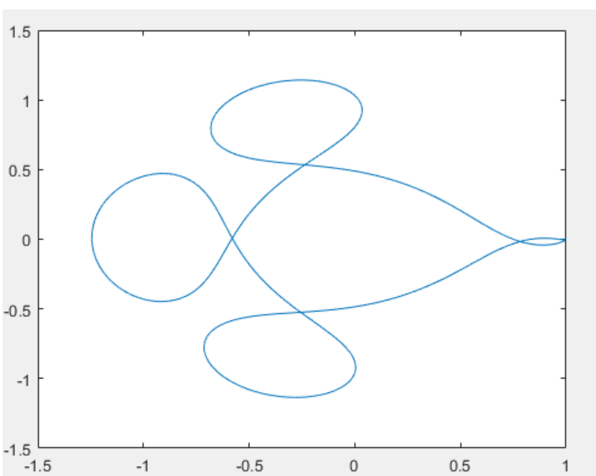
100



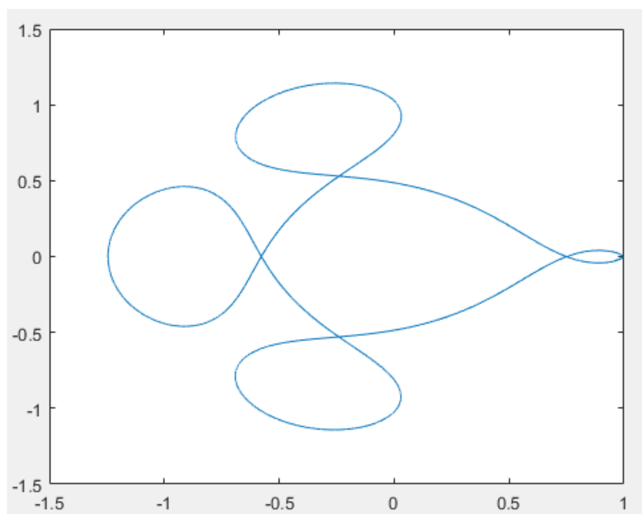
1000



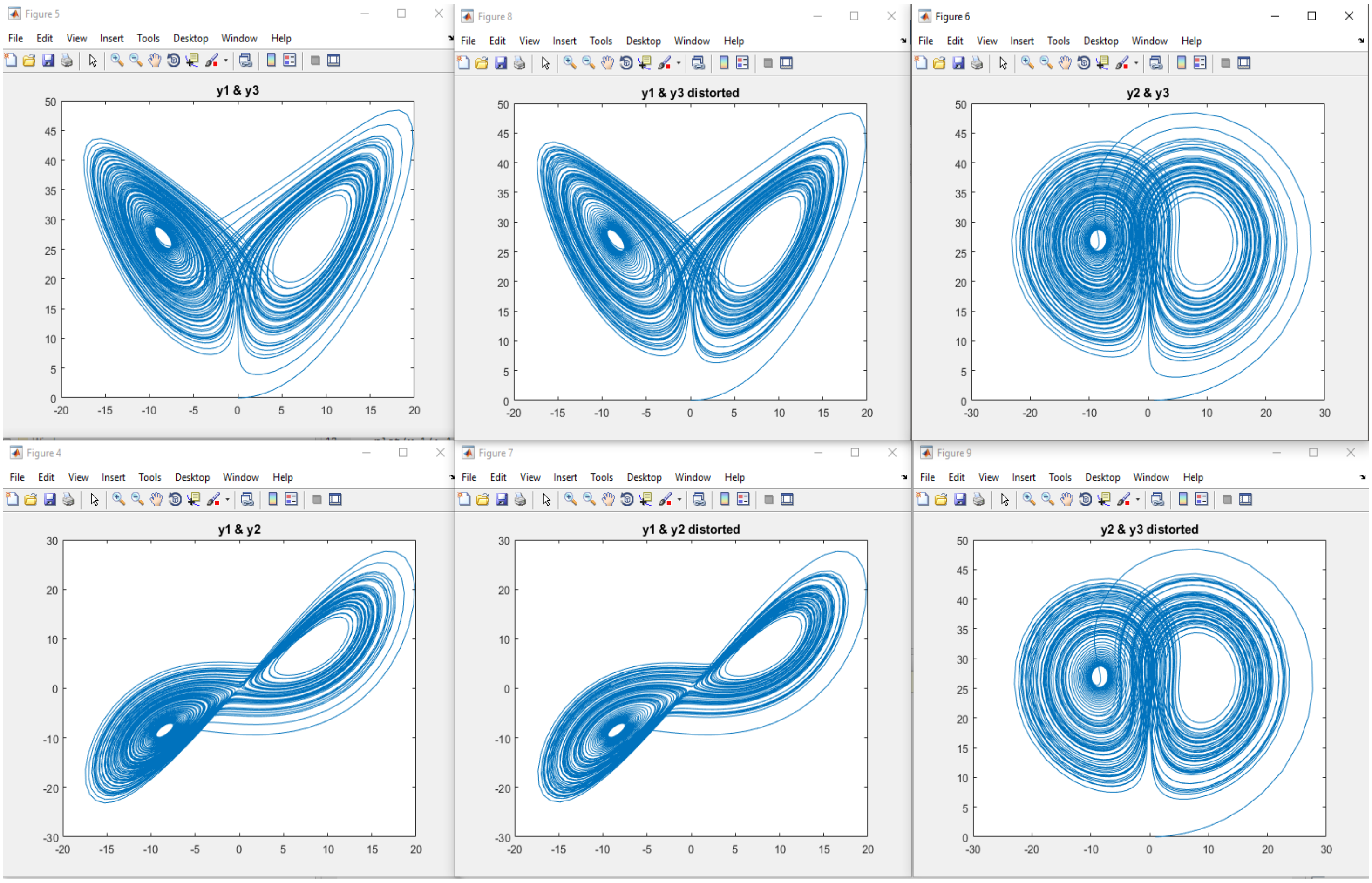
10 000

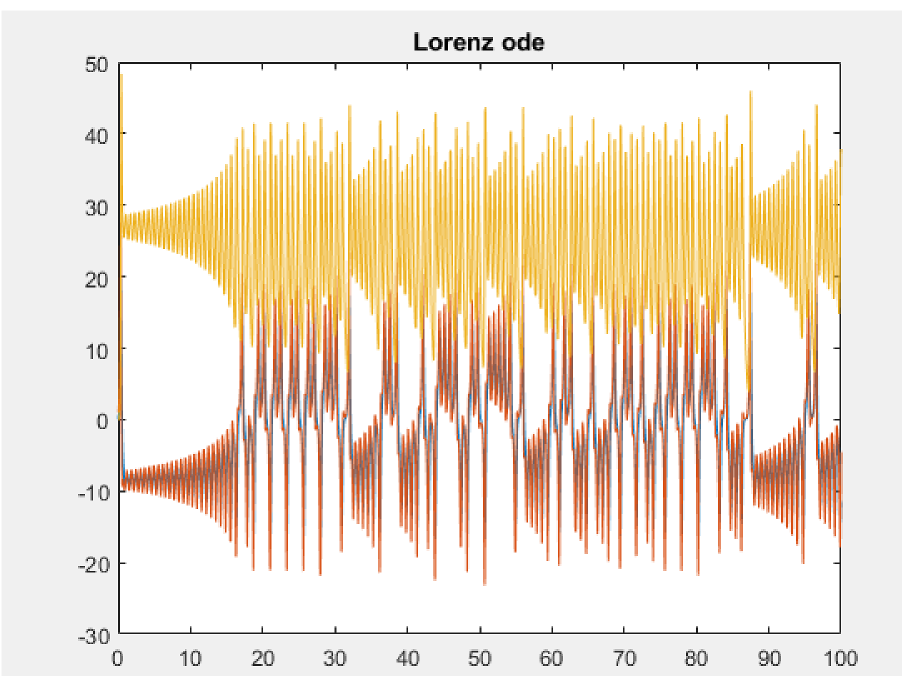


20 000



4. The differences in the following graphs by changing the initial values by 1e-10 varies more and more the further into computed solutions of the solution. You can see the change in the slightly distorted graphs by comparing the ends in each of them. There is a greater variance in the ends plots





```
% Question 1 for Assignment 4 4x03
x = 1.92:h:2.08; n = 9;
h = (2.08-1.92)/161;
f = @(x) ((x-2).^9);
fun1 = f(x);fun2 = zeros(1,161);
pnts = 1;
a = [-512 2304 -4608 5376 -4032 2016 -672 144 -18 1];
b = zeros(1,n+1);
for i = x
    b(n+1) = a(n+1);
    for j = n:-1:1
        b(1,j) = a(1,j) + i*b(1,(j+1));
    end
    fun2(1,pnts) = b(1,1);
    pnts = pnts + 1;
end

figure;
plot(x,fun1); title('[x-2]^9');
figure;
plot(x,fun2); title('Horners function');

f3 =@(x) x.*(x.*(x.*(x.*(x.*(x.*(x.*(x.*(x - 18) + 144) - 672) + 2016) - 4032) + 5376) -
4608) + 2304) - 512;
[a,b,c,d, var, r] = Bisec(fun2, 1.92, 2.08, 100, 1e-6); r
error = abs(r-2)

fd =@(x) x.^9 - 18.*x.^8 + 144.*x.^7 - 672.*x.^6 + 2016.*x.^5 - 4032.*x.^4 + 5376.*x.^3 -
4608.*x.^2 + 2304.*x - 512;
fsolve(fd,1.9)
function [a,b,fa,fb, var, c] = Bisec(f,a,b,max,eps)
    a0 = a;
    h = (b-a)/161;
    fa = f(1);
    fb = f((b - a0)/h);
    error = b-a;var = 0;
    for n = 0:max
        error = error/2;
        c = a+error;
        fc = f(round((c - a0)/h));
        var = var + 1;
        if abs(error) < eps
            fprintf('Conv, %i %i', n, error);
            return;
        end
        if sign(fa)~=sign(fc)
            b = c; fb = fc;
        else
            a = c;fa = fc;
        end
    end
end



% Question 2 for Assignment 4 4x03
```

```
close all; clear all;
tol = 1e-6;

x_a=15; y_a=-2;
X_a=[x_a;y_a];
i=0;err=1;
while(err > tol && i<100)
    x_a=X_a(1);y_a=X_a(2);
    Jac=[1 -3*y_a^2+10*y_a-2; 1 3*y_a^2+2-14]; % hand calc
    Fa= [x_a-y_a^3+5*y_a^2-2*y_a-13; x_a+y_a^3+y_a^2-14*y_a-29]; % computed each
iteration
    H=Jac\Fa;
    err=norm(H);
    X_a=X_a-H;
    i=i+1;
end
options=optimset('Display','off');
fun=@myfun1;
init=[15, -2];
[final,fval]=fsolve(fun,init,options);
fprintf('fsolve: %.3d,%.3d\n', final(1),final(2));
fprintf('Newt: %.3d,%.3d\n', x_a,y_a);
fprintf('i: %i\n\n', i);

x_b=(1+sqrt(3))/2; y_b=(1-sqrt(3))/2; z_b=sqrt(3);
X_b=[x_b;y_b;z_b];
i=0; err=1;
while(err>tol && i<100)
    x_b=X_b(1);y_b=X_b(2);z_b=X_b(3);
    Jac=[2*x_b 2*y_b 2*z_b; 1 1 0; 1 0 1]; % Jacobian computed by hand
    Fb= [x_b^2+y_b^2+z_b^2-5; x_b+y_b-1;x_b+z_b-3]; % computed each iteration
    H=Jac\Fb;
    err=norm(H);
    X_b=X_b-H;
    i=i+1;
end
options=optimset('Display','off');
fun=@myfun2;
init=[(1+sqrt(3))/2, (1-sqrt(3))/2, sqrt(3)];
[final,fval]=fsolve(fun,init,options);
fprintf('fsolve: %.3d ,%.3d, %.3d \n', final(1),final(2),final(3));
fprintf('Newt: %.3d, %.3d, %.3d\n', x_b,y_b,z_b);
fprintf('i: %i\n\n', i);

x_c=1;y_c=2;z_c=1;k_c=1;
X_c=[x_c;y_c;z_c;k_c];
i=0;err=1;
while(err>tol && i<100)
    x_c=X_c(1);y_c=X_c(2);z_c=X_c(3); k_c=X_c(4);
    Jac=[1 10 0 0;0 0 sqrt(5) -sqrt(5);0 2*y_c-2*z_c 2*z_c-2*y_c 0; ...
        2*sqrt(10)*(x_c-k_c) 0 0 2*sqrt(10)*(k_c-x_c)]; % computed by hand
    Fc= [x_c+10*y_c; sqrt(5)*(z_c-k_c);(y_c-z_c)^2;sqrt(10)*(x_c-k_c)^2];
    H=Jac\Fc;
    err=norm(H);
    X_c=X_c-H;
    i=i+1;
end
options=optimset('Display','off');
fun=@myfun3;
```

```
init=[1 2 1 1];
[final,fval]=fsolve(fun,init,options);
fprintf('fsolve: %d ,%d, %d, %d \n', final(1),final(2),final(3),final(4));
fprintf('Newt: %d, %d, %d, %d\n', x_c,y_c,z_c,k_c);
fprintf('i: %i\n\n', i);

x_d=1.8; y_d=0;
X_d=[x_d;y_d];
i=0; err=1;
while(err >tol && i<100)
    x_d=X_d(1);y_d=X_d(2);
    Jac=[1 0; 10/(x_d+0.1)-10*x_d/(x_d+0.1)^2 4*y_d];
    Fd= [x_d; 10*x_d/(x_d+0.1)+2*y_d^2];
    H=Jac\Fd;
    err=norm(H);
    X_d=X_d-H;
    i=i+1;
end
options=optimset('Display','off');
fun=@myfun4;
init=[1.8, 0];
[final,fval]=fsolve(fun,init,options);
fprintf('fsolve: %d,%.3d\n', final(1),final(2));
fprintf('Newt: %d,%.3d\n', x_d,y_d);
fprintf('i: %i\n\n', i);

x_e=0; y_e=0;
X_e=[x_e;y_e];
i=0;
err=1;
while(err >tol && i<100)
    x_e=X_e(1);y_e=X_e(2);
    Jac=[1e4*y_e 1e4*x_e; -exp(-x_e) -exp(-y_e)];
    Fe= [1e4*x_e*y_e-1; exp(-x_e)+exp(-y_e)-1.0001]; % computed each iteration
    H=Jac\Fe;
    err=norm(H);
    X_e=X_e-H;
    i=i+1;
end
options=optimset('Display','off');
fun=@myfun5;
init=[0, 0];
[final,fval]=fsolve(fun,init,options);
fprintf('fsolve: %d,%.3d\n', final(1),final(2));
fprintf('Newt: %d,%.3d\n', x_e,y_e);
fprintf('i: %i\n\n', i);


function F1 = myfun1(x)
    F1 = [x(1)+x(2)*(x(2).*(5-x(2))-2) - 13;
          x(1)+x(2)*(x(2).*(1+x(2))-14) - 29];

end

function F2 = myfun2(x)
    F2 = [x(1).^2+x(2).^2+x(3).^2;
          x(1)+x(2)-1;
          x(1)+x(3)-3];
end
```

```
function F3 = myfun3(x)
    F3 = [x(1)+10*x(2);
         sqrt(5)*(x(3)-x(4));
         (x(2)-x(3))^2;
         sqrt(10)*(x(1)-x(4))^2];
end

function F4 = myfun4(x)
        F4 = [x(1);
        10.*x(1)/(x(1)+0.1)+2.*x(2).^2];
end

function F5 = myfun5(x)
        F5 = [10.^4.*x(1).*x(2) - 1;
        exp(-x(1))+exp(-x(2))-1.0001];
end


% Question 3 for Assignment 4 4x03
n = 10000;
a = 0; b = 17.1;
h = (b-a)/n;
mu = 0.012277471;
muh = 1-mu;
f1 = @(t,y1,y2,y3,y4) y2;
f2 = @(t,y1,y2,y3,y4) y1 + 2*y4 - muh*((y1+mu)/(((y1+mu)^2 + y3^2)^(3/2))) - mu*((y1-
muh)/(((y1-muh)^2 + y3^2)^(3/2)));
f3 = @(t,y1,y2,y3,y4) y4;
f4 = @(t,y1,y2,y3,y4) y3 - 2*y2 - muh*(y3/(((y1+mu)^2 + y3^2)^(3/2))) - mu*(y3/(((y1-
muh)^2 + y3^2)^(3/2)));

t = zeros(1,n); t_a = a;
u1 = zeros(1,n); u1_0 = 0.994;
u2 = zeros(1,n); u2_0 = 0;
u3 = zeros(1,n); u3_0 = 0;
u4 = zeros(1,n); u4_0 = -2.00158510637908252240537862224;

t(1) = t_a;
u1(1) = u1_0;
u2(1) = u2_0;
u3(1) = u3_0;
u4(1) = u4_0;
for j = 1:n
    k1 = h*f1(t(j),u1(j),u2(j),u3(j),u4(j)); l1 = h*f2(t(j),u1(j),u2(j),u3(j),u4(j));
    m1 = h*f3(t(j),u1(j),u2(j),u3(j),u4(j)); n1 = h*f4(t(j),u1(j),u2(j),u3(j),u4(j));
    k2 = h*f1(t(j)+0.5*h,u1(j)+0.5*k1,u2(j)+0.5*l1,u3(j)+0.5*m1,u4(j)+0.5*n1); l2 =
h*f2(t(j)+0.5*h,u1(j)+0.5*k1,u2(j)+0.5*l1,u3(j)+0.5*m1,u4(j)+0.5*n1);
    m2 = h*f3(t(j)+0.5*h,u1(j)+0.5*k1,u2(j)+0.5*l1,u3(j)+0.5*m1,u4(j)+0.5*n1); n2 =
h*f4(t(j)+0.5*h,u1(j)+0.5*k1,u2(j)+0.5*l1,u3(j)+0.5*m1,u4(j)+0.5*n1);
    k3 = h*f1(t(j)+0.5*h,u1(j)+0.5*k2,u2(j)+0.5*l2,u3(j)+0.5*m2,u4(j)+0.5*n2); l3 =
h*f2(t(j)+0.5*h,u1(j)+0.5*k2,u2(j)+0.5*l2,u3(j)+0.5*m2,u4(j)+0.5*n2);
    m3 = h*f3(t(j)+0.5*h,u1(j)+0.5*k2,u2(j)+0.5*l2,u3(j)+0.5*m2,u4(j)+0.5*n2); n3 =
h*f4(t(j)+0.5*h,u1(j)+0.5*k2,u2(j)+0.5*l2,u3(j)+0.5*m2,u4(j)+0.5*n2);
    k4 = h*f1(t(j)+h,u1(j)+k3,u2(j)+l3,u3(j)+m3,u4(j)+n3); l4 =
h*f2(t(j)+h,u1(j)+k3,u2(j)+l3,u3(j)+m3,u4(j)+n3);
    m4 = h*f3(t(j)+h,u1(j)+k3,u2(j)+l3,u3(j)+m3,u4(j)+n3); n4 =
h*f4(t(j)+h,u1(j)+k3,u2(j)+l3,u3(j)+m3,u4(j)+n3);
```

```
    t(j+1) = t(j) + j*h;
    u1(j+1) = u1(j) + (1/6)*(k1+2*k2+2*k3+k4);
    u2(j+1) = u2(j) + (1/6)*(l1+2*l2+2*l3+l4);
    u3(j+1) = u3(j) + (1/6)*(m1+2*m2+2*m3+m4);
    u4(j+1) = u4(j) + (1/6)*(n1+2*n2+2*n3+n4);
end
plot(u1,u3);


% Question 4 for Assignment 4 4x03

[t,y]= ode45(@dydt,[0 100],[0;1 ;0]);
[t_1,y_1]= ode45(@dydt,[0 100],[1e-10;1+1e-10 ;1e-10]);
plot(t,y(:,1),t,y(:,2),t,y(:,3))
title('Lorenz ode');

figure;plot(y(:,1),y(:,2));title('y1 & y2');
figure;plot(y_1(:,1),y_1(:,2));title('y1 & y2 distorted');
figure;plot(y(:,1),y(:,3));title('y1 & y3');
figure;plot(y_1(:,1),y_1(:,3));title('y1 & y3 distorted');
figure;plot(y(:,2),y(:,3));title('y2 & y3');
figure;plot(y_1(:,2),y_1(:,3));title('y2 & y3 distorted');

function odefun=dydt(t,y)
odefun=zeros(3,1);
sig = 10;
r = 28;
b = 8/3;
odefun(1)= sig*(y(2)-y(1));
odefun(2)= r*y(1)-y(2)-y(1)*y(3);
odefun(3)= y(1)*y(2)-(b)*y(3);
end
```